An Adaptation of Experimental Design to the Empirical Validation of Software Engineering Theories

N. Juristo, A.M. Moreno

Facultad de Informática - Universidad Politécnica de Madrid -
Campus de Montegancedo s/n, 28660 Madrid
Tel.: + 34 91 336 69 22; Fax: + 34 91 336 69 17
{natalia, ammoreno}@fi.upm.es

Abstract

This paper has two objectives. Firstly, it seeks to promote discussion and debate about the need to encourage experimentation of the claims in the field of software engineering. The software community's lack of concern for the need for the aforesaid experimentation is slowing down adoption of new technology by organizations unfurnished with objective data that show the benefits of the new artifacts to be introduced. This situation is also leading the introduction of new software technology to be considered as a risk, because, as it has not been formally validated beforehand, its application can cause disasters in user organizations. The second objective is to present a formal method of experimentation in SE, based on the experimental design and analysis techniques used in other branches of science.

1. Introduction

Companies are continuously developing new, increasingly complex and, ultimately, more expensive software systems. This should be a condition for applying the range of development artifacts in a reliable manner. Paradoxically, however, real-world developments are often used as a culture medium for validating these artifacts, with the ensuing risks. There is no denying, unfortunately, that the models and theories outputted by Software Engineering (SE) research are not checked against reality as often as would be necessary to assure their validity for use in software construction. This can lead to justified distrust when applying the new solutions developed at laboratories or research centers in industry.

It is, therefore, essential to apply a process of experimental testing to validate any contribution made to SE. This paper seeks to highlight the need for an empirical validation of all artifacts used in SE, and then proposes an approach to introduce this based on experimental design techniques, widely used in other fields of science and engineering. Other researchers, including Basili [Basili, 86] and Pfleeger [Pfleeger, 95], have published work on experimental design and SE. In this paper, we aim to address in detail particular points, such as the parameters to be controlled in a SE experiment, and will set out several examples of how different types of experimental design can be applied to SE.

So as show the lack of empirical validation in the field of SE, the authors have compared what we have called *the essence of the scientific method* with SE research. The essence of the scientific method relates to certain characteristics common to the different methods of research with regard to the manner of attaining new knowledge. These common features can be divided into the following activities:

- **Interaction with reality**, which involves obtaining facts from reality. It can be performed by means of observation, where researchers merely perceive facts from the outside, or by means of experimentation, where researchers subject the object to new conditions and observe the reactions.
- **Speculation**, where researchers think about the perception obtained from the outside world. The results of this thinking range from a mere description of particular cases, through hypotheses and models, to general laws and theories.
- **Checking ideas against reality** in order to assure the truth of the speculations. It can safely be said that it is this stage that lends research its scientific value, as the stages of interacting with reality and speculation occur in other intellectual disciplines far from being considered scientific; for example, philosophy, religion, politics, etc. A branch of human knowledge attains the status of scientific when speculations are verifiable and, therefore, valid (although this status is always held provisionally until contradicted by a new reality). Remember that engineering fields depend on scientific knowledge to

build their artifacts.

When comparing the essence of the scientific method and research in SE, there are a series of discrepancies, including importantly the lack of emphasis on the experimental validation activity. In fact, present scientific progress in the software community appears to be based on natural selection. That is, researchers throw their lucubrations into the arena almost untested. After a few years or decades, theoretically, the fittest survives. Note the risk involved in this manner of scientific progress, as fashion, researcher credibility, etc., also play a prominent role in science. This way of selecting valid knowledge involves important risks when industry applies this new knowledge.

Statements claiming that SE experimentation is not needed can be heard frequently in SE. One of the arguments is that the "Romans built bridges and were not acquainted with the scientific method". Obviously, humans can generate valid knowledge by means of trial and error. However, this approach is longer and more chancy than the scientific method. If a critical software system fails and causes a disaster, could we say that we in SE prefer the old trial-and-error approach rather than experimental validation as called for by the scientific method? Another justification used to refute SE experimentation is based on trusting in intuition. Several examples can be used to reject this statement, for example, the fact that small software components are proportionally less reliable than larger ones, as reported by Basili [Basili, 94] among others. In [Tichy, 98] the author presents some arguments traditionally used to reject the usefulness of experimentation in this area with the corresponding refutation.

Although there are some experimental studies in the computer science literature [Prechelt, 98] [Frankl, 93] [Seaman, 98] [Iyer, 90], this is not the general rule. The want of experimental rigor in SE has already been stressed by authors like Zelkowitz [Zelkowitz, 98] or Tichy [Tichy, 93] [Tichy, 95], who base this affirmation on a study of the papers published in several system-oriented journals. Surveys such as Zelkowitz's and Tichy's tend to validate the conclusion that the SE community can do a better job in reporting its results, making them more trustworthy and thus making it easier for industry to adopt the new research results.

2. Experimental Design for Software Engineering

Once that the need for empirical validation in SE has been assumed, the authors propose an approach to introduce it based on experimental design techniques [Box, 78] [Selwyn, 96] [Clarke, 97] [Edwards, 98] used in others fields of science.

Empirical validation can be carried out in several situations : *laboratory* validation of theories, validation at the level of *real projects* and validation by means of *historical data*. Unlike the other two methods, laboratory validation allows greater control of the different parameters that affect software development. Real projects allow data considered to be relevant for the study in question to be collected. Validation using historical data allows researchers to work with data on finished projects, employing the most relevant for the experiment to be conducted. Zelkowitz [Zelkowitz, 98] and Kitchenham [Kitchenham, 96] suggested similar classifications. Zelkowitz groups experimental approaches into three broad categories: *controlled methods, observational methods* and *historical methods*, while Kitchenham refers to these categories of experimentation as *formal experiments*, *case studies*, and *surveys*. An example of experimentation with real projects is the experience factory proposed by Basili [Basili, 95], historical data have been applied by McGarry [McGarry, 97] among others, and formal experiments have been studied by Pfleeger [Pfleeger, 95] in the DESMET project.

In this paper, we focus on formal experiments and present an in-depth study of the application of experimental design to SE empirical validation, placing special emphasis on the adaptation of experimental design terminology to SE. Table 1 summarizes the above-mentioned experimentation process. Table 2 describes the application of experimental design concepts to SE. Table 3 shows the value of some of the experimental design concepts for SE experimentation. Finally, Table 4 presents a summary of the experimental design techniques that can be applied.

Phase of the experiment	Description
Defining the Objectives of the Experiment.	The mathematical techniques of experimental design demand that experiments produce quantitative results. Therefore formal experimentation in SE requires quantifiable hypotheses. This hypothesis will be usually expressed in terms of a metric of the software product developed using the software artifact to be analyzed or of the development process where this artifact has been applied.
Designing the Experiment	In order to plan experimentation in SE according to experimental design guidelines, its terminology has to be applied to SE. See table 2 with the terminology employed in experimental design for generic experimentation, and its application to experiments in SE. The next step is to select the experimental design technique. This technique will determine how many experiments are required, how many times each experiment has to be repeated and what data we need to output to ascertain the validity of the conclusions. There are different techniques of experimental design depending on the aim of the experiment, the number of factors, the levels of the factors, etc. Table 4 shows a brief summary of the most commonly used experimental design techniques.
Executing Experiments	The software engineer is now ready to execute the experiments indicated as a result of the preceding design stage, measuring the response variables at the end of each experiment.
Analyzing Results	This stage is also called Experimental Analysis. The software engineer will quantify the impact of each factor and each interaction between factors on the variation of the response variable. This is what is referred to (according to experimental design terminology) as "the statistical significance of the differences in the response variable due to the different levels of each factor". • If there is no statistical significance, the variation in the response variable can be put down to chance or to another variable not considered in the experiment. • If there is statistical significance, the variation in the response variable is due to the fact that a certain level (or combination of levels of different factors) causes improvements in the response variable. When we have understood the impact, we can ascertain which alternative of which factor significantly improves the value of the response variable. Depending on the experimental design technique applied in the preceding stage, a different statistical technique must be used to achieve the above objective. This is not the place to expound the underlying mathematics of experimental analysis. Interested readers are referred to the references already mentioned. Section 3 shows some examples of SE experiments illustrating different experimental design and analysis techniques.

Table 1. Phases of the Experimental Design Process used for SE Experiments

Concept	Description	Application in SE
Experimental unit	Entity used to conduct the experiment	Software projects
Parameters	Characteristic (qualitative or quantitative) of the experimental unit	See table 3
Response variable	Datum to be measured during the experimental unit	See table 3. Note there are no response variables relating to the "problem". This is because response variables are data that can be measured *a posteriori*, that is, once the experiment is complete. In the case of SE, the experiment involves development (in full or in part) of a software system to which particular technologies are applied. The characteristics of the *problem* to be solved are the experiment input data, that is, they stipulate how it will be performed. As such, they are parameters and factors of the experiment. However, they are not experimental output data that can be measured and, thus, do not generate response variables.
Factor	Parameter that affects the response variable and whose impact is of interest for the study	Factors are chosen from the parameters in table 3. Factors have different values during the experiment
Level	Possible values or alternatives of the factors	Values of factors in table 3
Interaction	The effect of one factor depends on the level of another	Relations between the parameters in table 3; for example, problem complexity and product complexity
Replication	Repetition of each experiment to be sure of the measurement taken of the response variable	Repeatability in SE must be based on analogy, not on identity; the different experiments will consist of similar problems, similar processes, similar teams, etc.
Design	Specification of the number of experiments, selection of factors, combinations of levels of each factor for each experiment and the number of replications per experiment	The design will indicate the number of software projects, factors and their alternatives that will be used during experimentation, as well as the number of replications of the experiments, based on analogy.

Table 2. Application of experimental design concepts to SE

PARAMETERS			
PROBLEM (User need)	PROCESSES of construction employed	PERSONS (team of developers)	PRODUCT
– Definition (poorly/well defined problem) – Need volatility (very/hardly/non volatile need) – Ease of understanding (problem well/poorly/fairly well understood by developers) – Problem complexity – Problem type (data processing, knowledge use, etc.), – Problem-solving type (procedural, heuristic, real-time problem solving, etc.) – Domain (aeronautics, insurance, etc.) – User type (expert, novice, etc.)	– Maturity – Description (set of phases, activities, products, etc.) – Relationship between members (definition of interrelations between team members) – Automation (in which phases or activities tools are used) – Risks	– Number of members – Division by positions (no. of software engineers, programmers, project managers, etc.) – Years of experience of each member in development – Experience of each member in the problem type – Experience of each member in the software process applied – Background of each member (discipline of origin) – Type of relationship between members (all in the same building, same town, subcontracts, etc.)	– Type of life cycle to be followed – Software type (OO, databases, real time, expert system, etc.) – Size – Complexity – Architecture/Organization – Hardware platform – Interaction with other software – Processing conditions (batch, on-line, etc.) – Security requirements – Response-time requirements – Documentation required – Help required
RESPONSE VARIABLES			
PROBLEM	PROCESS	PERSONS	PRODUCT
	– Schedule deviation – Budget deviation – Compliance with construction process – Products obtained (do they comply with the process stipulations?)	– Productivity – User satisfaction – usability – usefulness	– Correctness of products obtained (no. of errors, etc.) – Validity of the products (compliance with customer expectations) – Portability, Maintainability, Extendibility, Performance, Flexibility, Interoperability,…

Table 3. Proposal of Parameters and Response Variables for SE research

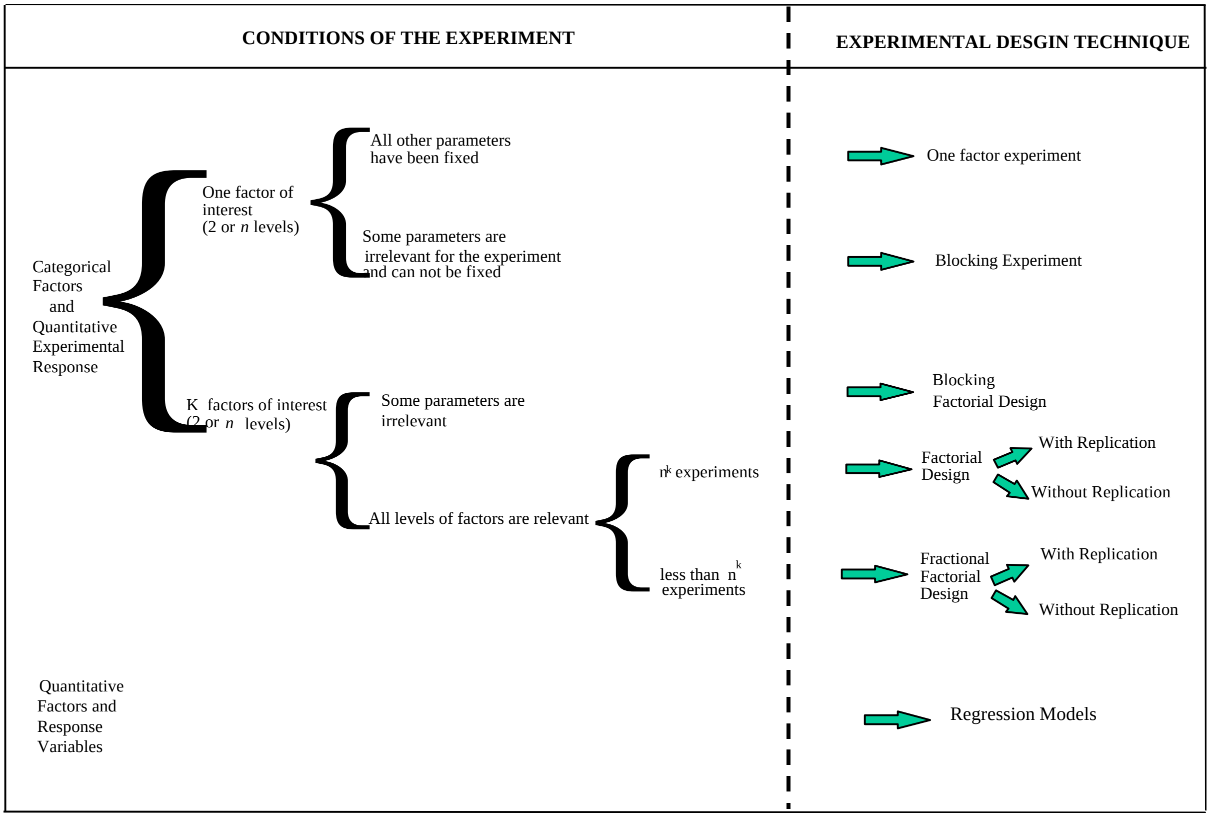
CONDITIONS OF THE EXPERIMENT
EXPERIMENTAL DESGIN TECHNIQUE
Categorical Factors and Quantitative Experimental Response
One factor of interest (2 or *n* levels)
All other parameters have been fixed
Some parameters are irrelevant for the experiment and can not be fixed
K factors of interest (2 or *n* levels)
Some parameters are irrelevant
All levels of factors are relevant
n^k experiments
less than n^k experiments
Quantitative Factors and Response Variables
One factor experiment
Blocking Experiment
Blocking Factorial Design
Factorial Design
With Replication
Without Replication
Fractional Factorial Design
With Replication
Without Replication
Regression Models

Table 4. Different Experimental Design Techniques

3. Example of SE Experiments using Experimental Design

This section presents two examples of possible SE experiments employing the experimental design process described in Table 1. Depending on the experimental desgin techinque used, different analysis methods must be applied. During the experimental analysis phase, we will not enter into a detailed justification of all the mathematical calculations; our objective is simply to give readers a taste of what sort of work could be performed during an experimentation in SE, avoiding the tiresome, though simple, calculations called for by experimental analysis.

3.1. One Factor Experiment

Suppose we are researching on a CASE tool, and we think it will increase programmers productivity. We will compare this tool with two other tools widely used in industry and each experiment will be repeated five times, in order to consider experimental errors. The response variable will be *programmers productivity* (lines of code/person-day) and all other parameters of table 3 will be fixed. This is an example of one factor experiment. This kind of experimental design is used to determine the best choice of k alternatives (in our case of three alternatives).

Table 5 shows the fifteen observations of the response variable (column Z contains the values for the new tool).

R	V	Z
144	101	130
120	144	180
176	211	141
288	288	374
144	72	302

Table 5. Value of the response variables

The analysis if this experiment is shown in table 6. From this table we can know that the mean value of productuvity of a CASE tool is 187,7 lines/person-day. The effects of tools R, V and Z are -13,3, -24,5 and 37,7, respectively. That means that tool R provides 13,3 lines less than the mean, tool V provides 24,5 lines less than the mean, and tool V provides 37,7 lines more than the mean.

	R	V	Z	
	144	101	130	
	120	144	180	
	176	211	141	
	288	288	374	
	144	72	302	
Sum of the column	$\sum Y_{\bullet 1} = 872$	$\sum Y_{\bullet 2} = 816$	$\sum Y_{\bullet 3} = 1127$	$\sum Y_{\bullet\bullet} = 2815$
Mean of the column	$\overline{Y}_{\bullet 1} = 174.4$	$\overline{Y}_{\bullet 2} = 163.2$	$\overline{Y}_{\bullet 3} = 225.4$	$m\overline{Y}_{\bullet\bullet} = 187.7$
Effect of the column	$a_1 = \overline{Y}_{\bullet 1} - \overline{Y}_{\bullet\bullet} = -13.3$	$a_2 \overline{Y}_{\bullet 2} - \overline{Y}_{\bullet\bullet} = -24.4$	$a_3 \overline{Y}_{\bullet 3} - \overline{Y}_{\bullet\bullet} = 37.7$	

Table 6. Data from the experimental analysis of the example

The second step involves calculating the sum of the squared errors (SSE) in order to estimate the variance of the errors and the confidence interval for effects. For that aim each observation will be divided in three parts: the grand mean, the effect of the tool, and the residuals. For each part we have used a matrix notation.

$$\begin{bmatrix} 144 & 101 & 130 \\ 120 & 144 & 180 \\ 176 & 211 & 141 \\ 288 & 288 & 374 \\ 144 & 72 & 302 \end{bmatrix} = \begin{bmatrix} 187.7 & 187.7 & 187.7 \\ 187.7 & & \\ 187.7 & & \\ 187.7 & & \\ 187.7 & & 187.7 \end{bmatrix} + \begin{bmatrix} -13.3 & -24.5 & 37.7 \\ & & \\ & & \\ & & \\ -13.3 & -24.5 & 37.7 \end{bmatrix} + \begin{bmatrix} -30.4 & -62.2 & -95.4 \\ -54.4 & -19.2 & -45.4 \\ 1.6 & 47.8 & -84.4 \\ 113.6 & 124.8 & 148.6 \\ -30.4 & -91.2 & 76.6 \end{bmatrix}$$

SSE = $\sum_{i=1}^{r}\sum_{j=1}^{a}$ e_{ij}^2 = (-30,4)^2+ (-54,4)^2+ ... +(76,6)^2 = 94.365,20

Next step is calculating the variation in the response variable due to the factor and to the experimental error. For that aim we calculate the sum of squares total (SST).

SST = r$\sum_{j}\ {}_{j}^{2}$ + SSE = 5 ((-13,3)^2 + (-24,5)^2 + (37,6)^2) + 94.365,2 = 105.357,3

The percentage of variation in the response variable explained by CASE tools is 10,4% (10.992,13/105.357,3). The rest of the variation 89,6% is due to experimental errors. That means that the experiment has not been planned properly.

In order to determine whether the variation of 10,4% in the productivity has statistical significance we have to use the ANOVA (Analysis Of VAriance) technique, with the F-test function and table (this table is not included in the paper, readers can find them in the bibliography of experimental design mentioned above). The technique seeks to compare the contribution of the factor to the variation in the response variable with the contribution of the errors. If the variation due to errors is high, a factor that explains a high variation in the response variable might has not statistical significance. In order to determine the statistical significance we will compare the computed F-value with the value got from the F-table, as shown in table 7.

Table 8 shows the ANOVA analysis for our example. The calculated F-value is smaller than the one got from the F-table. Therefore, we can, again, conclude that the difference in productivity is mainly due to experimental errors instead of to the CASE tools. In that sense, we can state that neither tool provides more productivity than the others.

COMPONENT	SUM OF SQUARES	PERCENTAGE OF VARIATION	DEGREES OF FREEDOM	MEAN SQUARE	F-COMPUTED	F-TABLE
Y	$SSY = \sum Y_{ij}^2$		ar			
$\bar{Y}_{\bullet\bullet}$	$SSO = arm^2$		1			
$Y - \bar{Y}_{\bullet\bullet}$	$SST = SSY - SSO$	100	ar-1			
A	$SSA = r\sum a_i^2$	$100\left(\frac{SSA}{SST}\right)$	a-1	$MSA = \frac{SSA}{a-1}$	$\frac{MSA}{MSE}$	$F_{[1-\alpha;a-1,a(r-1)]}$
e	$SSE = SST - SSA$	$100\left(\frac{SSE}{SST}\right)$	a(r-1)	$MSE = \frac{SSE}{a(r-1)}$		

$$S_e = \sqrt{MSE}$$

Table 7. ANOVA table for one factor experiments

Y	633,639.00					
Y..	528,281.69					
Y-Y..	105.357,31	100.00	14			
A	10.992,13	10.4	2	5496.1	0.7	2.8
Errors	94.365,20	89.6	12	7863.8		

$$S_e = \sqrt{MSE} = \sqrt{7863.77} = 88.68$$

Table 8. ANOVA table for our experiment

3.2. Factorial Design with Replication

Suppose that we have invented a new development paradigm that is completely different from the structured and OO paradigms and want to confirm that our innovation improves development projects. We will centre on *correctness* as the response variable, measured, for example, by the number of faults emerging three months after software deployment. There are a lot of characteristics that have an impact on this response variable: problem complexity, problem type, process maturity, team experience, software complexity, integration with other software, etc. However, all of these will be fixed at an intermediate value (that is, they will be selected as parameters of the experiment), except *development paradigm,* and *software complexity* which will be factors. Each factor will necessarily admit two alternatives to simplify the calculations. According to experimental design guidelines, the factors, labelled with letters, and their alternatives, labelled with level 1 and -1, are listed, as shown in table 9.

FACTOR	NAME	LEVEL -1	LEVEL 1
Paradigm	A	New	OO
Software complexity	B	Complex	Simple

Table 9. Factors and levels of the experiment

We will use a factorial design with replication as all levels of our factors are relevant for the experiment, and we want to consider the experimental errors. In order to evaluate the experimental errors we will repeat each experiment three times, so we will get twelve measurements of the response variable.

Taking the measurements of the response variable and the values assigned to the factors in table 9, the first step of the experimental analysis is to build what is called the sign table. As shown in table 10, the first column of the matrix is labelled I, and it contains all 1s. The next two columns, labelled with the factor names, contain all the possible combinations of -1 and 1. The fourth column is the product of the entries in columns A and B. The twelve observations are then listed in column Y. The entries in column I are then multiplied by those in last column, and the sum is then entered under column I. The entries in column A are then multiplied by those in last column and the sum is entered under column A. This column multiplication operation is repeated for the remaining columns in the matrix. The sum under each column is divided by 4 to give the corresponding coefficients of the regression model.

I	A	B	AC	Y	Mean $\overline{Y}$
1	-1	-1	1	(15, 18, 12)	15
1	1	-1	-1	(45, 48, 51)	48
1	-1	1	-1	(25, 28, 19)	24
1	1	1	1	(75,75,81)	77
164	86	38	20		Total
41	21.5	9.5	5		Total/4

Table 10. Sign table for a 2^2 experimentation with replication

The second step involves calculating SSE. Table 11 shows the estimated response and the errors for each of the twelve observations. The estimated value for the response variable is calculated adding the products of the effects (C_0, C_A, C_B, C_{AB}) and the entries (X_A, X_B, X_{AB}) in the sign table.

Effects					Estimated Response	Mean Response			Errors		
i	I 41	A 21.5	B 9.5	AB 5	$\hat{Y}_i$	Y_{i1}	Y_{12}	Y_{i3}	e_{i1}	e_{i2}	e_{i3}
1	1	-1	-1	1	15	15	18	12	0	3	-3
2	1	1	-1	-1	48	45	48	51	-3	0	3
3	1	-1	1	-1	24	25	28	19	1	4	-5
4	1	1	1	1	77	75	75	81	-2	-2	4

Table 11. Errors in each experiment

The sum the squared errors is:

$$SSE = \sum_{i,j} e_{i,j}^{\ 2} = 0^2+3^2+(-3)^2+(-3)^2+ 0^2+3^2+1^2+(-5)^2+(-2)^2+(-2)^2+4^2 = 102$$

Now we want to calculate the variation in the response variable due to each factor or combination of factors, and to the experimental error. For that aim we calculate SST.

$$SST = 2^2 r\, C_A^{\ 2} + 2^2 r\, C_B^{\ 2} + 2^2 r\, C_{AB}^{\ 2} + \sum_{i,j} e_{i,j}^{\ 2} = 5{,}547 + 1{,}083 + 300 + 102 = 7{,}032$$

Factor A explains 78,88% (5,547/7,032) of the variation, factor B explains 15,04% and the interaction AB explains 4,27%. The rest of the variation, 1,45%, is a variation non explicated, and therefore, due to experimental errors.

4. Conclusions

In this paper, we presented a possible adaptation of the experimental design techniques used in other branches of science and engineering to perform experiments in SE.

The objective of the paper is not only to present a means of carrying out formal experimentation in SE but also to promote discussion and debate on the need to encourage experimentation of the claims in this field. The software community’s lack of concern for the need for the aforesaid experimentation is slowing down adoption of new technology by organizations unfurnished with objective data that show the benefits of the new artifacts to be introduced. This situation is also leading the introduction of new software technology to be considered as a risk, because, as it has not been formally validated beforehand, its application can cause disasters in user organizations.

We are aware that software development's marked economic and commercial nature can be a decisive factor standing in the way of the necessary experimentation, as experimentation does not produce tangible, short-term benefits. The benefit of experimentation will come to fruition in future development projects, and this benefit is difficult to quantify at the time of deciding on experimental feasibility or the number of experiments to be performed. However, as we have already said, experimentation can also stop industry taking unnecessary risks by adopting proposals that have not been satisfactorily tested.

5. References

[Basili, 84] V.R. Basili, B.T. Perricone. *Software Errors and Complexity: An Empirical Investigation*. **Communications of the ACM**, January 1984, pp. 42-52.

[Basili, 86] V.R. Basili, R.W. Selby, D.H. Hutchens. *Experimentation in Software Engineering*. **IEEE Transactions on Software Engineering**, vol. 12 (7), July 1986, pp. 733-743.

[Basili, 95] V. R. Basili. *The Experience Factory and Its Relationship to Other Quality Approaches*, Academic Press Inc., **Adnvances in Computers**, Volume 41, 1995.

[Box, 78] Box, G.E.P., Hunter W.G. and Hunter, J.S. **Statistics for Experiments**. Wiley, New York, (USA), 1978.

[Clarke, 97] Clarke, G.M. and Kempson, R.E. **Introduction to the Design & Analysis of Experiments**. Wiley & Sons, New York (USA), 1997.

[Edwards, 98] Edwards, A.L. **Experimental Design**. Addison-Wesley Educational Publishers, Delaware (USA), 1998.

[Frankl, 93] P.G. Frankl, S.N. Weiss. *An Experimental Comparison of the Effectiveness of Branch Testing and Data Flow Testing*. **IEEE Transactions on Software Engineering**, vol. 19 (8), August 1993.

[Iyer, 90] Iyer, R.K. *Special Section on Experimental Computer Science*. **IEEE Transactions on Software Engineering**, vol. 16 (2), February 1990.

[Kitchenham, 96] Kitchenham, B. *Evaluating Software Engineering Methods and Tools*. Parts 1 to 8. **SIGSOFT** Notes 1996 and 1997.

[McGarry, 97] F. McGarry, S. Burke, W. Deker and J. Haskell. *Measuring Impacts of Software Process Maturity in a Production Environment*. **22nd NASA Workshop on Software Engineering**, Maryland, USA, December 1997, pp. 193-220.

[Pfleeger, 95] Pfleeger, S.L. *Experimental Design and Analysis in Software Engineering*. **Annals of Software Engineering**, vol. 1, 1995, 219-253.

[Prechelt, 98] Prechelt, L and Tichy, W.F. *A Controlled Experiment to Assess the Benefits of Procedure Argument Type Checking*. **IEEE Transactions on Software Engineering**, vol. 24 (4), April 1998, 302-318.

[Seaman, 98] Seaman, C.B. and V.R. Basili. *Communication and Organization: An Empirical Study of Discussion in Inspection Meetings*. **IEEE Transactions on Software Engineering**, vol. 24 (7), July 1998, 559-572.

[Selwyn, 96] Selwyn, M.R. **Principles of Experimental Design for the Life Sciences**. CRC Press Inc. (UK) 1996.

[Tichy, 93] Tichy, W.F. *On Experimental Computer Science*. *International Workshop on Experimental Software Engineering Issues. Critical Assessment and Future Directions*. Proceedings, 1993, 30-32.

[Tichy, 95] Tichy, W.F. et al. *Experimental Evaluation in Computer Science: A Quantitative Study*. **Journal of Systems and Software**, vol. 28, 1995, 9-18.

[Tichy, 98] Tichy, W.F. *Should Computer Scientists Experiment More ?* **IEEE Computer**, May 1998,32-40.

[Zelkowitz, 98] Zelkowitz, M, Wallace, R. *Experimental Models for Validating Technology*. **IEEE Computer**, May 1998, 23-31.